AFRL-IF-RS-TR-2001-223
Final Technical Report
October 2001

DENIAL OF SERVICE (DOS) ATTACK ASSESSMENT ANALYSIS REPORT

Johns Hopkins University

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. J754/01

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2001-223 has been reviewed and is approved for publication.

APPROVED: 

FRANK S. LAMONICA
Project Engineer

FOR THE DIRECTOR: MICHAEL L. TALBERT, Technical Advisor
Information Technology Division
Information Directorate

If your address has changed or if you wish to be removed from the Air Force Research Laboratory Rome Research Site mailing list, or if the addressee is no longer employed by your organization, please notify AFRL/IFTD, 525 Brooks Rd, Rome, NY 13441-4505. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	Oct 01	Final Apr 00 - Jun 01

4. TITLE AND SUBTITLE
DENIAL OF SERVICE (DOS) ATTACK ASSESSMENT ANALYSIS REPORT

5. FUNDING NUMBERS
C - F30602-00-C-0070
PE - 63760E
PR - IAST
TA - 00
WU - 08

6. AUTHOR(S)
D.M. Gregg, W.J. Blackert, D.C. Furnanage, D.V. Heinbuch

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Johns Hopkins University
Applied Physics Laboratory
11100 Johns Hopkins Road
Laurel, MD 20723-6099

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Advanced Research Projects Agency
3701 Fairfax Drive
Arlington, VA 22203-1714

AFRL/IFTD
525 Brooks Rd
Rome NY 13441-4505

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2001-223

11. SUPPLEMENTARY NOTES
AFRL Project Engineer: Frank S. Lamonica, IFTD, 315-330-2055

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
This effort applied modeling and simulation techniques to the analysis of denial of service attacks on a target network. Attack models were quantified and mitigation techniques were examined and analyzed with possible adaptations of the attacker. The analyses revealed that attack classes and their dependence on network topology can be identified.

14. SUBJECT TERMS
Denial of Service, Information Security, Modeling and Simulation

15. NUMBER OF PAGES
72

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

TABLE OF CONTENTS

Appendix

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF TABLES

Section 1

INTRODUCTION

The Denial of Service Attack Assessment (DOSAA) project, funded by the Fault-Tolerant Network (FTN) program, has modeled and validated five denial of service (DoS) attacks:

a. Octopus

b. Address Resolution Protocol (ARP) Cache Poison

c. Transmission Control Protocol (TCP) SYN Flood [Distributed Denial of Service (DDoS)]

d. Snork

e. User Datagram Protocol (UDP) Storm

These attack models were executed against a validated model of a target network whose architecture and stochastic behavior was varied for the analysis. OPNET Modeler, a commercial communications network modeling and simulation package, was used to support model development.

Section 2

BACKGROUND

2.1 ATTACK MODELS

Table 2-1 summarizes the attack models produced and validated (Reference 1) under the DOSAA project. Sections 3 through 7 describe each attack in more detail.

Table 2-1 DOSAA Attack Models

Attack	Affected Layer	Description
Octopus	Application	Open as many connections as possible on a remote host to invoke executables for each request
ARP Cache Poison	Data Link	Causes Internet Protocol (IP) address to map to invalid Ethernet address
TCP SYN Flood (DDoS)	Transport	Distributed TCP SYN Flood attack; fills server pending connection queue
Snork	Transport	UDP packet used to consume resources by inducing continuous bounce of packets between systems
UDP Storm	Transport	Exploit services to flood the network

2.2 NETWORK MODEL DEVELOPMENT

For DOSAA, The Johns Hopkins University Applied Physics Laboratory (JHU/APL) produced a target network model of an existing JHU/APL subnet. This target model (or a variation) was used to examine the effects of the five attacks. Figure 2-1 depicts the topology of the JHU/APL subnet model. The subnet (and therefore the model) has 104 nodes: 82 are Windows NT clients, 11 are UNIX, and 11 are printers. Services supported on the computer systems include email, File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP), Rlogin, and Telnet. Most workstations within the subnet are connected over copper in a star topology via a Cisco switch. Connected to this configuration is the Information Operations (IO) Laboratory local area network (LAN). Within the IO Laboratory is a variety of NT, Windows, and UNIX workstations connected using a hub, as well as a second group of workstations (behind a Gauntlet firewall) connected via a second hub. The backbone in the IO Laboratory is a 10-MB Ethernet.

The Cisco switch connects to subnets throughout JHU/APL's enterprise network via other Cisco switches. Connected via one of these switches are three servers providing Web, email, Telnet, and FTP services to the 104-node subnet.

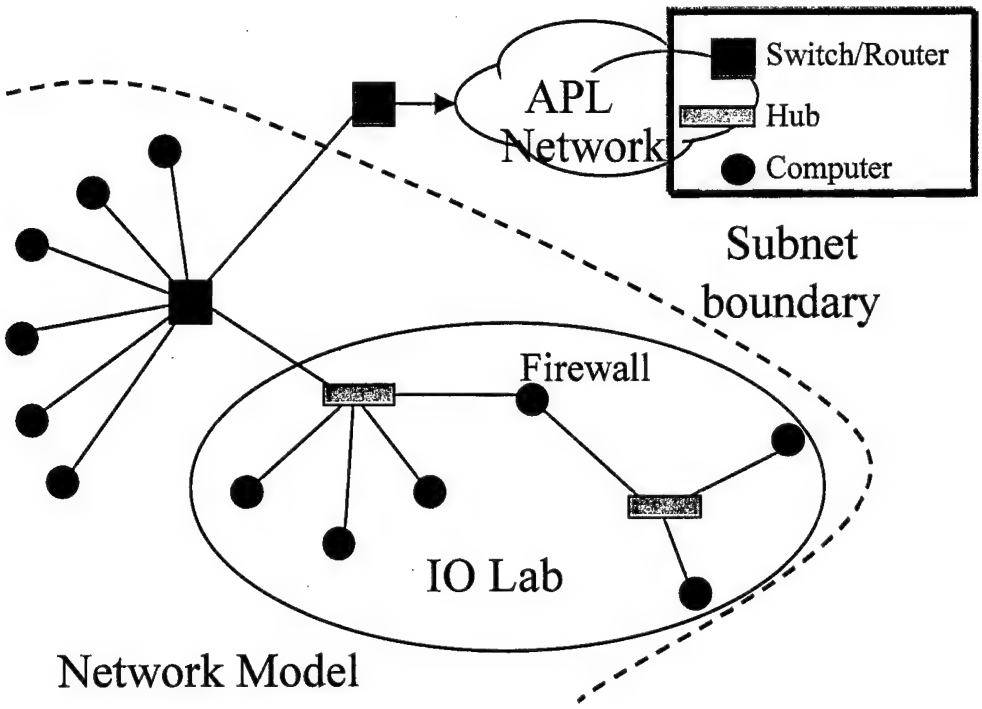

Figure 2-1 JHU/APL Subnet Topology

The OPNET model of the subnet uses a mix of standard OPNET Modeler models, as well as custom JHU/APL models. The standard node models include OPNET's generic switch, generic hub, server node supporting only printer services, and server node supporting an array of other services (email, Telnet, FTP, and HTTP). The protocol models include OPNET's model of IP and a modified version of the TCP model (JHU/APL has added the TCP SYN queue and connection limitations). The firewall model is also a JHU/APL-generated model that provides basic packet filtering.

Section 3

OCTOPUS ATTACK

This section contains analysis results for the Octopus attack, a stateful resource attack (Reference 2) where the attacker consumes resources at a server by requesting many services and connections in a very short period of time. Stateful resource attack behavior is driven by very specific parameters: the amount of resource to be consumed, the length of time the system holds those resources before releasing them, and the rate at which the attacker consumes resources. For the Octopus attack, these parameters are connection size, connection timeout, and the rate at which connection requests are received at the host.

3.1 NETWORK CONFIGURATION

The Octopus attack was examined in two network configurations. The first, shown in Figure 3-1, is derived from the testbed network used to validate the Octopus attack. The second configuration is the 104-node JHU/APL subnet model described in Subsection 2.2.

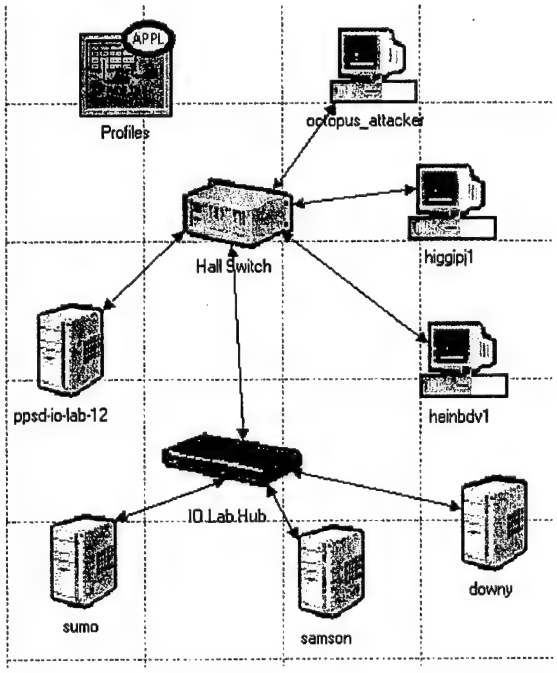

Figure 3-1 Small Network Configuration

3.2 ANALYSIS PARAMETERS

To examine the Octopus attack, the following observations were made, leading to the model runs defined in Table 3-1. Octopus functions by exhausting resources required for providing service. According to TCPdump data obtained when running Octopus against computers in the IO Laboratory, the victim client continues to successfully connect to the server under attack. However, instead of conducting a session, the connection is immediately torn down because no resources are available to spawn a child process. Because connections eventually time out at the server, the opportunity for a user to obtain service occurs in the interval between when an Octopus connection times out (thereby releasing some resources) and the time when the attacker generates another Octopus connection.

Table 3-1 Octopus Analysis Cases

Case	Timeout	Maximum Connections	Attack Start Time	Attack End Time
1	30	500	1000	2200
2	50	500	1000	2200
3	100	500	1000	2200
4	300	500	1000	2200
5	500	500	1000	3000
6	1000	500	1000	5000
7	300	500	1000	2200
8	300	1000	1000	2200
9	300	1500	1000	2200

The following parameters influence the ability of a user to acquire service while an Octopus attack is occurring:

a. Attack Interval – The opportunity for a client to acquire service decreases as the Octopus attack rate increases.

b. Connection Timeout – For a fixed-length attack, the shorter the connection timeout (i.e., the time between when an Octopus connection starts and when it is closed down), the more windows of opportunity exist for a client to acquire service.

c. Number of Supported Connections – At the beginning of an attack, there is a period of time between when no resources are being used and when all resources are used. This time, which is influenced by the number of supported connections and the rate at which Octopus connections are generated, is when legitimate users can still access the system. Once this period is exhausted, and assuming that the attack is continuous, there will be a period of time before any of the Octopus connections time out. When timeouts begin, the number of timeouts (which equates to the number of opportunities for legitimate service) is equal to the maximum number of connections.

Using these parameters as a basis, the model runs in Table 3-1 were used to examine Octopus attack effects.

Two measures of effectiveness (MOEs) are presented for the Octopus attack: (1) probability of denied service (PDS) and (2) outage time of a legitimate client during an Octopus attack. PDS and outage time are defined as follows:

PDS = 1 – (number of successful services ÷ number of service attempts)

Outage Time = (duration PDS > threshold) ÷ attack duration

3.3 OCTOPUS ANALYSIS RESULTS

Model results in Figure 3-2 show average PDS (averaged over 30 model runs) for varying timeout parameters. The model shows a clear dependency between attack effectiveness and the server timeout setting. Results show that decreasing the server timeout parameter lowers attack effectiveness. In fact, when the timeout parameter is set to 30 seconds, PDS is 0. It should be noted, however, that a 30-second timeout may have a secondary impact of denying service to legitimate users. In addition, as will be discussed, the attacker can counter the positive effects of lowering the timeout setting.

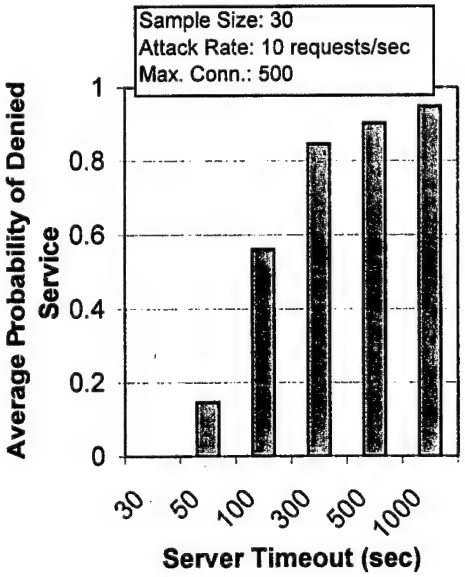

Figure 3-2 PDS as a Function of Server Timeout (Octopus Attack)

Attack effectiveness also decreases by increasing the total amount of the resource (i.e., maximum number of connections) at the server. These results are shown in Figure 3-3. In this case, PDS drops by close to 0.30 by increasing the server's maximum number of connections from 500 to 1500. However, there is a limit to the success of this approach because of the limit on a server's available resources (e.g., memory). The attacker can

counter the positive effects of this approach as well. Nevertheless, the model shows a clear dependency between attack effectiveness and maximum number of connections.

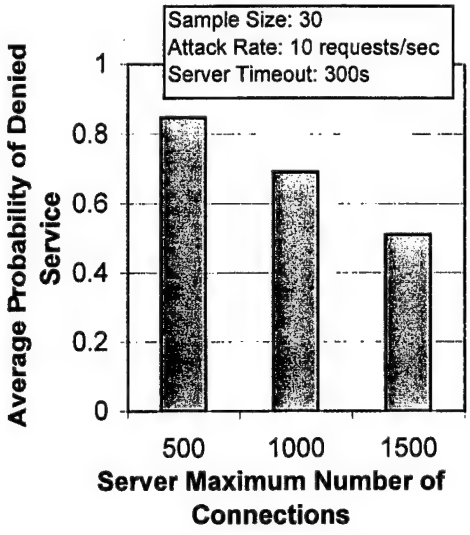

Figure 3-3 Average PDS as a Function of Maximum Number of Connections (Octopus Attack)

The results also show that the attacker can easily counter the impact of these adjustments by stepping up the rate at which resources are requested. Figure 3-4 shows average PDS as a function of attack rate for attack rates at 10, 20, and 100 requests per second. At 10 requests per second, PDS is below 0.2. However, as the attack rate increases, so does PDS, regardless of the 50-second server timeout setting.

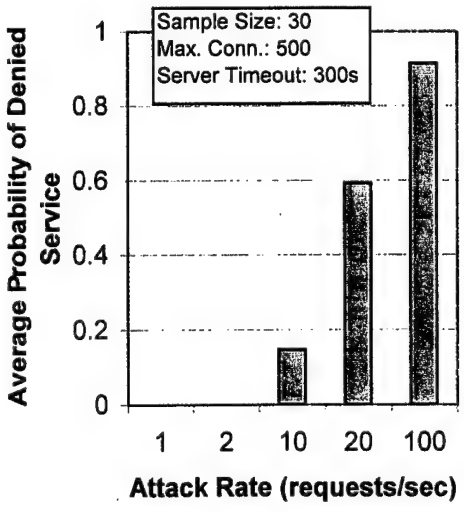

Figure 3-4 Average PDS as a Function of Attack Rate (Octopus Attack)

Interestingly, PDS results for the small network and 104-node network are not statistically different. This may suggest that the effectiveness of the Octopus attack is independent of network configuration, provided the client and attacker have comparable network resources.

Figures 3-5 and 3-6 provide outage duration results as a function of server timeout and maximum number of connections. Average PDS for these cases is shown at the top of both figures. For the cases with high average PDS, deviation in outage duration narrows between the three thresholds used.

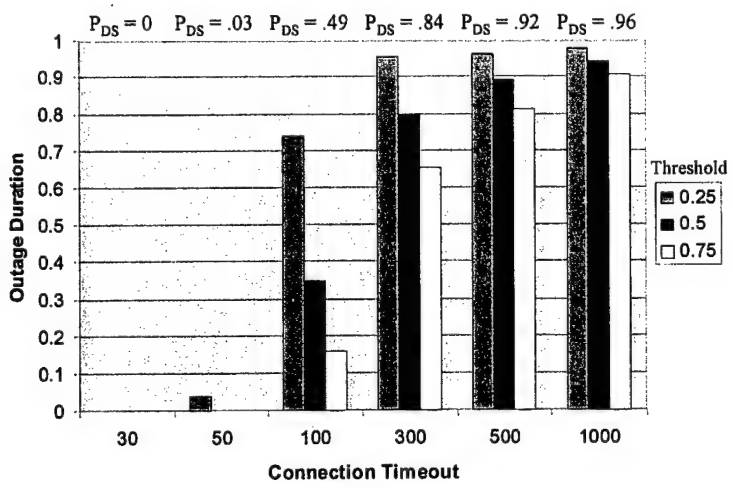

Figure 3-5 Outage Duration as a Function of Server Timeout (Octopus Attack)

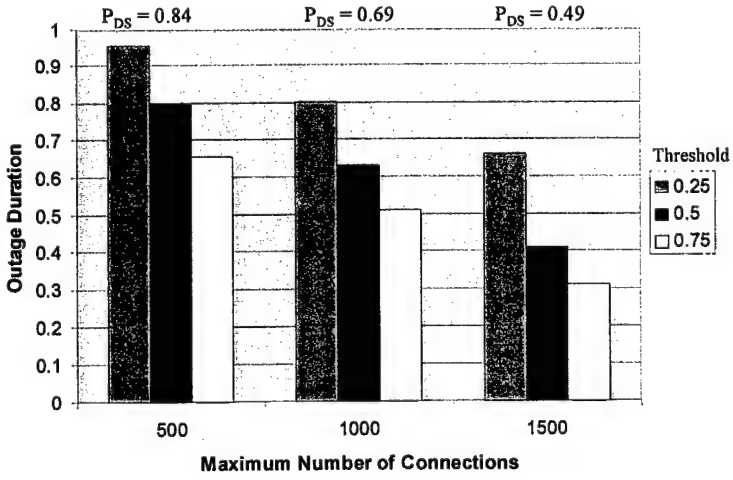

Figure 3-6 Outage Duration as a Function of Maximum Number of Connections (Octopus Attack)

Section 4

ARP CACHE POISON ATTACK

This section provides results for the ARP Cache Poison attack. A system uses the ARP to map IP addresses to hardware (e.g., Ethernet) addresses. A system maintains a local table, commonly referred to as an ARP cache, that contains IP address-hardware address pairs. When a system needs to perform a mapping, it first consults the ARP cache. If the IP address of interest is contained in the table, the mapping is performed locally. If the IP address is not contained in the table, an update to the table is necessary. A system updates the table by sending out an ARP request. The request is sent as a broadcast and is received by all systems on the subnet. These systems all update their local tables with the IP address and hardware address of the requesting system. The system that is using the hardware address in question replies to the ARP request. The ARP response is sent as a unicast to the requesting system. That system then updates its ARP table.

Each entry in an ARP table contains a timer parameter. This value is set to the number of seconds this entry should remain in the queue. ARP documentation suggests that these entries will time out, regardless of whether or not the entry is used. However, the common implementation suggests that each time an entry is used, its timeout value is reset.

ARP cache poisoning works by placing invalid ARP cache entries into a system's cache. Because the Ethernet has no packet delivery guarantee, a packet that uses the invalid Ethernet address is simply lost in the network. ARP caches are updated on the arrival of either a request or reply and therefore it is possible to poison the cache using either message. In the ARP attack studied, the attacker poisons the cache of a client by responding to a legitimate request made by the client. By poisoning the cache of the client with a false server hardware address, service is denied to that client.

4.1 INITIAL NETWORK CONFIGURATION

The OPNET model shown in Figure 4-1 was used for the initial analysis of the ARP Cache Poison attack. The network consists of a legitimate client, a legitimate server, and the attacker. The server provides HTTP service to the client. The attack is designed so that the attacker responds to ARP requests for the server's hardware address with a false hardware address (100). The client will send all information to the false address until either (1) a legitimate response is received, thereby overwriting the false information or (2) the client's cache clears out the entry (at which time, a new attacker/legitimate response race condition occurs).

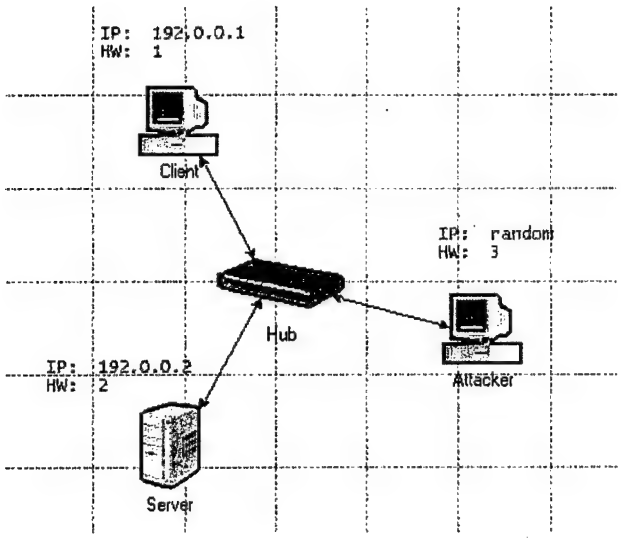

Figure 4-1 Network Model Used To Test ARP Cache Poison Attack

4.2 ANALYSIS PARAMETERS

For this analysis, one of the main goals was to study the effectiveness of the attack while varying the delay between when the attacker receives the request and when the poison response is sent. Other factors that determine the effectiveness of the attack include the server's method of updating the ARP cache and the frequency at which the cache is updated (ARP cache timeout value). For example, Windows NT 4.0 clears an ARP cache entry after 5 minutes without communication to the respective client. For the analysis performed here, the ARP cache timeout procedure was not varied, but the timeout value itself was varied. Because the update is dependent on client-server communication, the application request rate also becomes a factor as well as the ARP request-response network delay. Lastly, to deny service, the poisoned ARP packet must be received before the legitimate transaction between the client and server has transpired. The timing effects are shown in Figure 4-2, which depicts a timeline of legitimate network traffic as well as possible ARP poison attack times.

The following parameters are shown in Figure 4-2:

- T_{Rq} is the time at which the ARP request is sent from the server
- T_{Rs} is the time at which the ARP response is received by the server
- d is the network delay between when an ARP request is sent and when the response is received by the same node ($d = T_{Rs} - T_{Rq}$)
- P_1, P_2, and P_3 are different points in time that the poisoned ARP response is received
- A is the time is takes for a transaction between client and server to complete

- I is the application interarrival time defined by the frequency of application requests made by the client

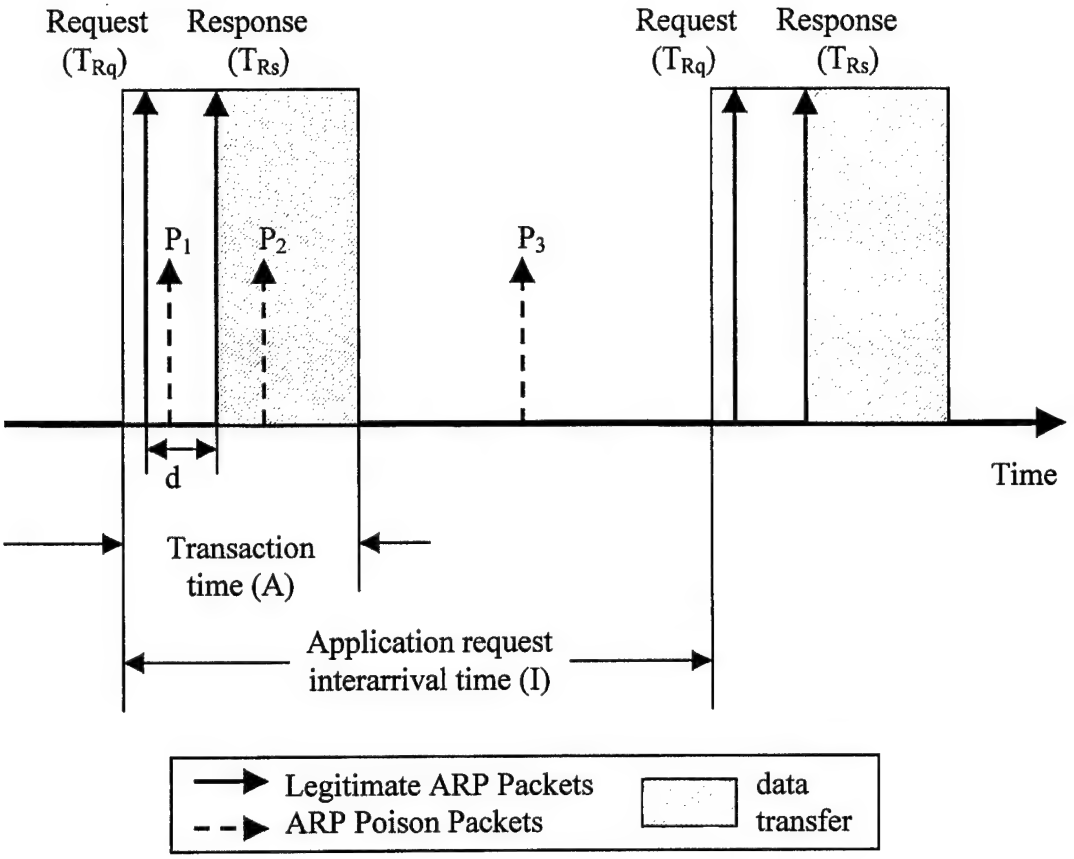

Figure 4-2 Timeline of Network Traffic

4.2.1 NETWORK DELAY

If the attacker sends packets such that they arrive at time P_1, the response is too soon (i.e., the ARP hardware address is overwritten by the legitimate one received at time T_{Rs}). If the poisoned packet arrives at time P_2, the attacker successfully denies service up until the transaction is complete. If the poisoned packet arrives at time P_3, the attacker does not deny service for the intended transaction, but provided the ARP cache entry does not timeout prior to the next transaction, the victim will be denied this next transaction. Thus, the attacker denies the client service if P is > T_{Rs} but occurs before the transaction time is over. The attacker does not deny service if P < T_{Rs}, a parameter that is defined by the network delay.

4.2.2 CACHE TIMEOUT AND APPLICATION INTERARRIVAL

Assuming a successful poison of the ARP cache of the victim, the victim will be denied service until the poisoned cache entry is updated by a legitimate ARP packet or dropped

b. Number of downloaded objects – The number of Web objects downloaded by a host as a function of time

4.3 ARP CACHE POISON ANALYSIS RESULTS

For the ARP Cache Poison attack analysis, the simulation was run for 12,000s and the attack was in place from 1000s to 10,000s. A baseline case (with no attack) was also run for each scenario (shown with dashed lines in the figures that follow). Only one dashed line appears because varying the poison delay and varying the ARP cache timeout do not affect the network when operating under no attack conditions. Results are obtained by averaging the data over 50 runs.

As expected, Figure 4-3 shows that attack effectiveness is dependent on the delay of the ARP cache poison. If the attacker responds immediately to the request (a 0-second delay), the order of arrival is dependant only on the network delay. As the attacker increases delay, the probability of the poison packet arriving after the legitimate response increases and the number of objects received begins to decrease. This is represented by the 0.2- and 0.5-msec lines (that overlap) and shows the minimum number of packets. However, as the delay continues to increase, the arrival time between the legitimate and poisoned responses continues to increase, resulting in more time for the client to send legitimate traffic. For this reason, the attack becomes less effective (more objects are downloaded).

Figure 4-4 shows that longer ARP cache timeouts (600s and 900s) lower the effectiveness of the attack (more objects are downloaded by the client). This is because any valid entries in the ARP cache must timeout before the attacker has a chance to poison the cache. Therefore, a longer timeout provides a longer period where legitimate connections can occur. Although the attack is less effective initially, these longer ARP cache timeouts result in attack effects being observable for a longer time after the attack completes (time > 10,000s). The results in Figure 4-4 assume a network model where the user is actively using the server when the attack begins. For this reason, there is a higher likelihood that valid connections will continue for some time.

Figures 4-5 and 4-6 show PDS for these same two parameters. These plots show the percent of all connections denied before, during, and after the attack. Figure 4-5 shows that attacker delay is an important factor in attack effectiveness. For attackers to optimize this type of attack, they must have knowledge of, or be able to guess, the delay within the network. Figure 4-6 shows that the ARP cache timeout can also have a significant effect on PDS, especially after the attack. In particular, the larger timeout values result in significant DoS after attack packets are no longer being sent.

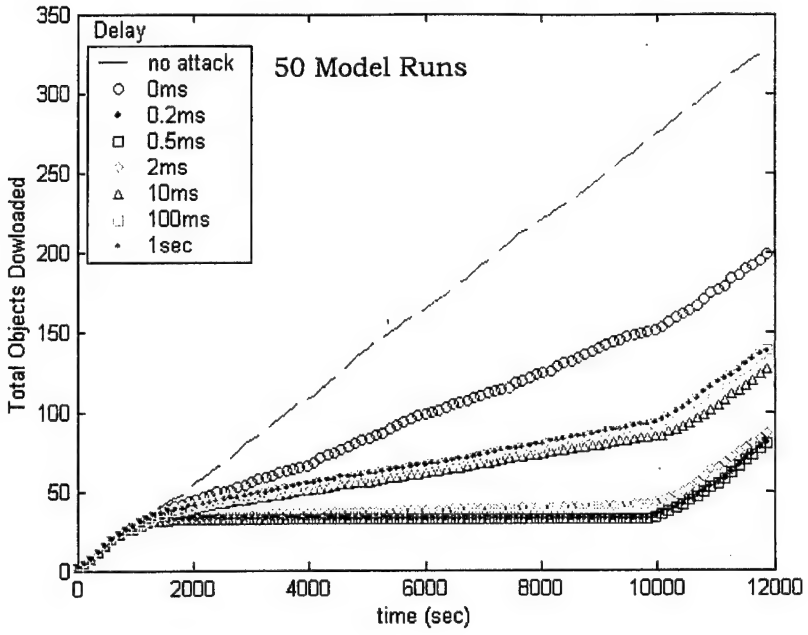

Figure 4-3 Average Number of Objects Downloaded by the Client as a Function of Attack Delay (ARP Cache Poison Attack)

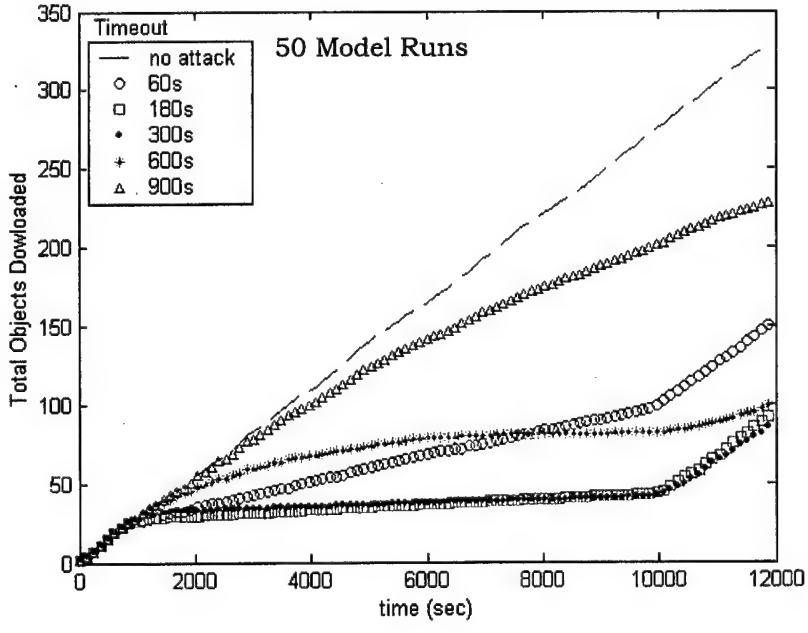

Figure 4-4 Average Number of Objects Downloaded by the Client for Different ARP Cache Timeouts (ARP Cache Poison Attack)

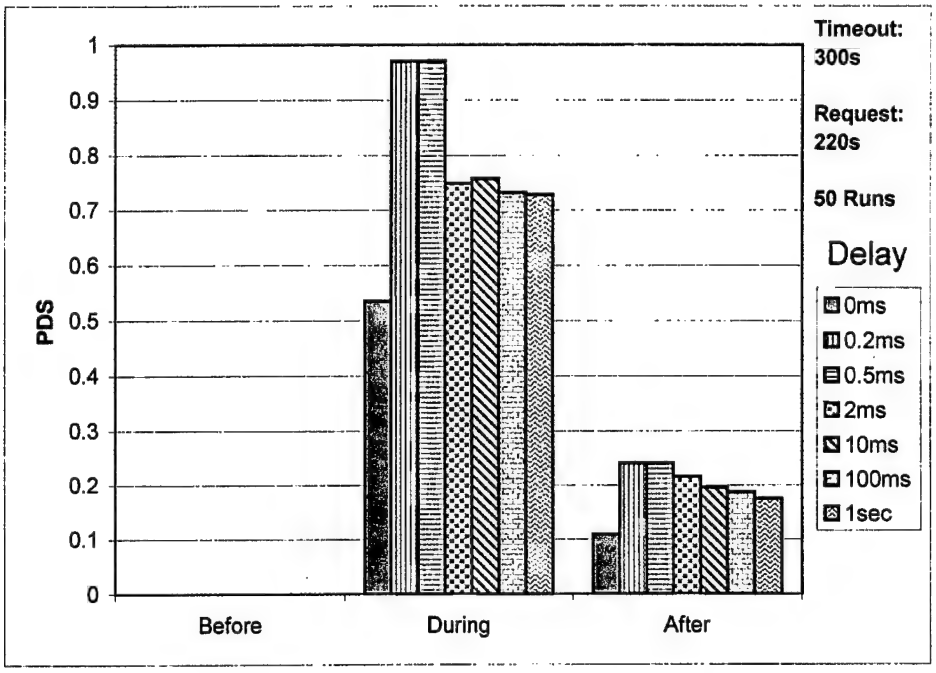

Figure 4-5 Average PDS as a Function of ARP Cache Poison Delay (ARP Cache Poison Attack)

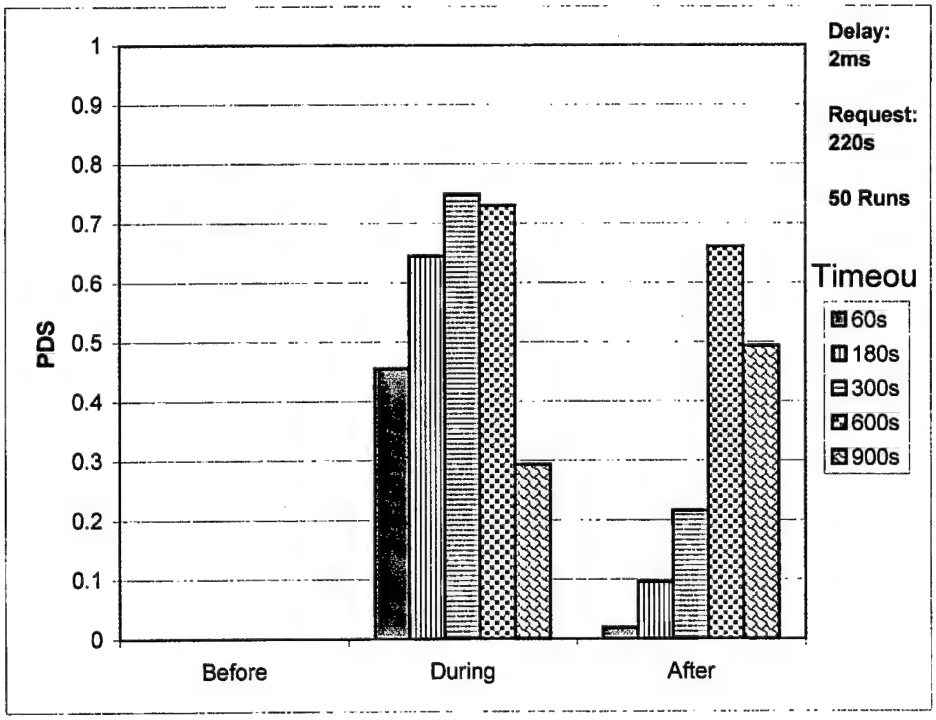

Figure 4-6 Average PDS as a Function of ARP Cache Timeout (ARP Cache Poison Attack)

To observe the effects of changing the network topology, the ARP cache poison attack model was executed against the 104-node target network model. Data was collected and averaged over 50 runs for 12,000s of simulation time (where the attack took place between 1000s and 10,000s). When varying the ARP cache timeout, results were consistent with the simple network tested. However, network topology did impact attack effectiveness for different attack delays. Figure 4-7 shows the number of objects downloaded; Figure 4-8 depicts PDS for different attacker delays.

For the more complex network, the attacker delay was varied from 0 msec to 1 sec (using 15 values within the range to determine the point at which the attacker becomes effective). Figure 4-7 shows the total objects downloaded by the client. The attacker is not effective (i.e., all of the baseline objects are downloaded) for delays between 0 msec and 0.11 msec. This differs from the smaller network tested where an attacker delay of 0 sec was able to decrease the number objects downloaded by nearly 50 percent. Similar to the simple network, delays greater than 0.5ms result in more objects downloaded (a less effective attack).

PDS results shown in Figure 4-8 also reflect a shift in the minimum effective attacker delay. For delays less than 0.13ms the probability that some portion of the service is denied is approximately zero. A 0.13ms delay resulted in PDS greater than 95%. These results are due to longer network delays in more complex networks than in simple networks. Therefore the attacker of a complex network must wait longer to send the ARP poison response in order to assure it is received after the legitimate response (thus overwriting the legitimate response). Although all delays between 0.13ms and 1sec result in some denial of service (PDS > 0), for both simple and complex network topologies, PDS decreases for the largest delays. In these cases, the ARP poison response is too late, arriving after the entire service has completed.

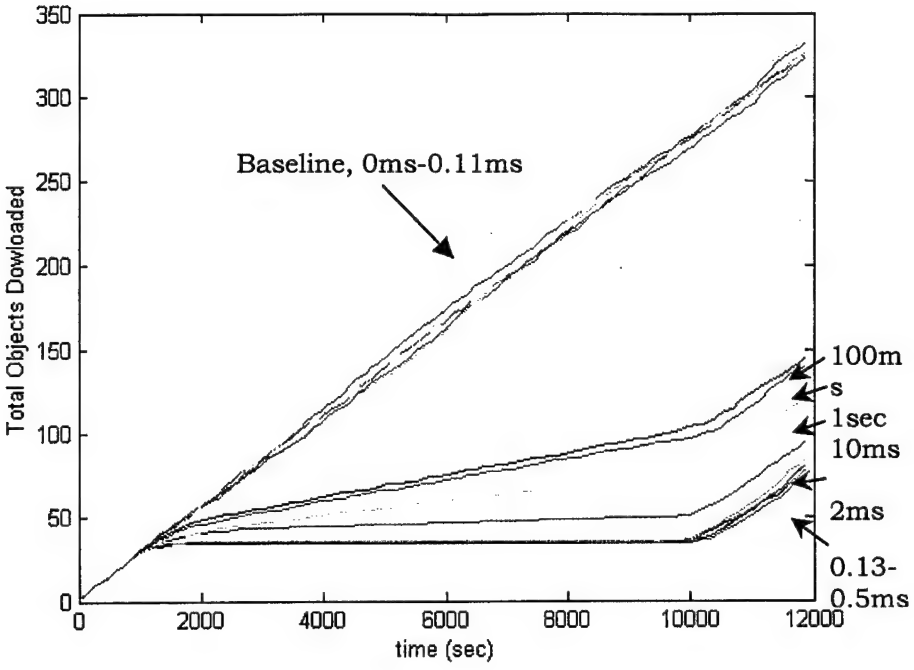

Figure 4-7 Average Number of Total Objects Downloaded by the Client for Different Attacker Delays (104-Node Network)

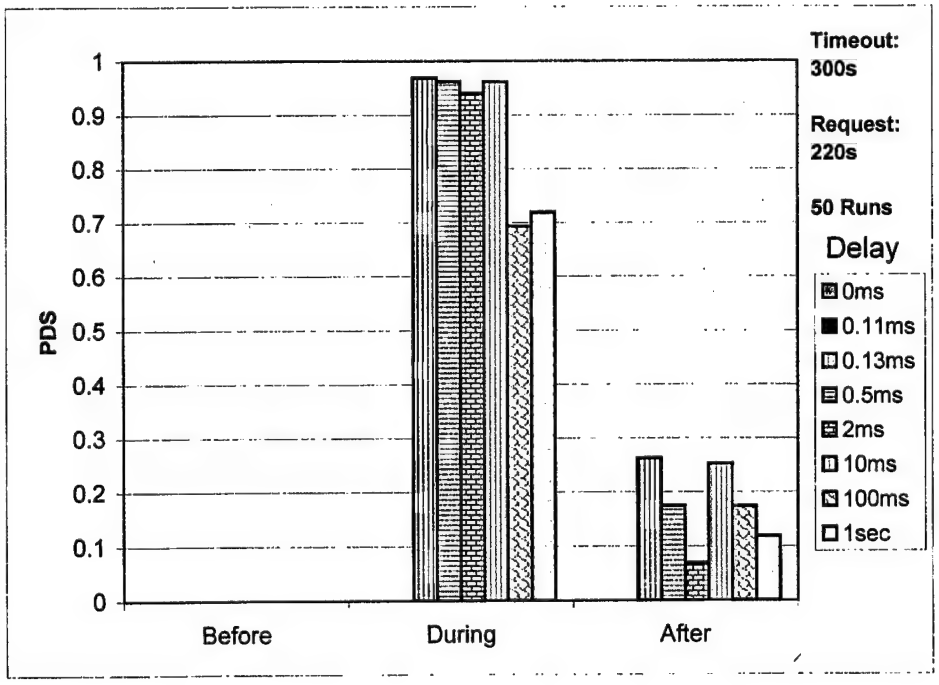

Figure 4-8 Average PDS as a Function of ARP Cache Poison Delay (104-Node Network)

Section 5

TCP SYN FLOOD (DDOS) ATTACK

This section provides analysis results for the TCP SYN Flood DDoS attack. A DDoS attack uses multiple entities (commonly referred to as zombies) to attack a system. Because each zombie attacks with all its resources, the aggregate attack effect can be more severe than an attack from a single host. DOSAA has investigated the TCP SYN flood variation of the DDoS as implemented in the Stacheldraht DDoS attack.

Like Octopus, the TCP SYN Flood attack is an instance of a stateful resource attack. Stateful resource attack behavior is driven by very specific parameters: the amount of resources to be consumed, the length of time the system holds the resources before releasing them, and the rate at which the attacker consumes resources. For the TCP SYN Flood attack, this translates into the pending connection queue size, timeout for queue entries, and rate at which SYN packets are received at the host.

5.1 NETWORK CONFIGURATION

To analyze the performance of the TCP SYN Flood attack, the JHU/APL subnet model was modified. First, the number of clients was reduced to half, while increasing each remaining client's traffic loading (this reduces memory usage in OPNET). Second, a small attack network was constructed containing the zombie systems. These zombie systems were based on systems used in the IO Laboratory for validation. The network is shown in Figure 5-1 (the JHU/APL subnet is contained with the Office Network icon).

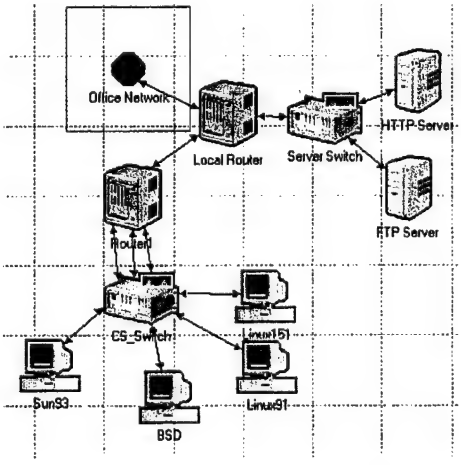

Figure 5-1 Network Layout for TCP SYN Flood Attack

5.2 ANALYSIS PARAMETERS

The TCP SYN Flood attack analysis examines the cases shown in Table 5-1. The default queue size of 1024 was selected because it is the queue size of current versions of Solaris (7 and 8). As stated previously, the subnet model was reduced to half the number of original nodes, while the default HTTP request interarrival time of 240 seconds was halved to 120 seconds to compensate. The attack SYN rate generated for validation was 5000 packets per second; that is used as a starting point in the analysis. HTTP is the service attacked.

Table 5-1 TCP SYN Flood Attack Analysis Cases

Case	Queue Size	Connection Queue Timeout (s)	Attack SYN Rate (pkps)
1	1024	15	5000
2	4096	15	5000
3	8192	15	5000
4	1024	90	5000
5	1024	180	5000
6	8192	15	10000

The MOEs selected for this analysis are as follows:

a. PDS – The probability that the connection queue is full when a legitimate packet arrives, causing the legitimate connection to be denied

b. Attack Duration – The amount of time the attack continues beyond the time the attacker ceases activity

c. Effect on Other Services – A measure of how other services are affected by the TCP SYN flood attack

5.3 TCP SYN FLOOD ATTACK ANALYSIS RESULTS

5.3.1 ANALYSIS OVERVIEW

For this attack, PDS = 1 – (number of successful connections ÷ total number of attempts) (e.g., the sum of the successful connections and the failed connections). In this case, failed connections occur because the pending connection queue is full when the SYN packet arrives for a new, legitimate connection. In calculating PDS, attack packets were not included in the totals.

Figure 5-2 presents PDS as a function of time for each analysis case (described in more detail later). In each case, PDS is calculated over a 10-second window and then the windowed data is averaged over all runs. As shown in the figure, service denials appear to continue for all cases beyond 700 seconds, when the attack terminates.

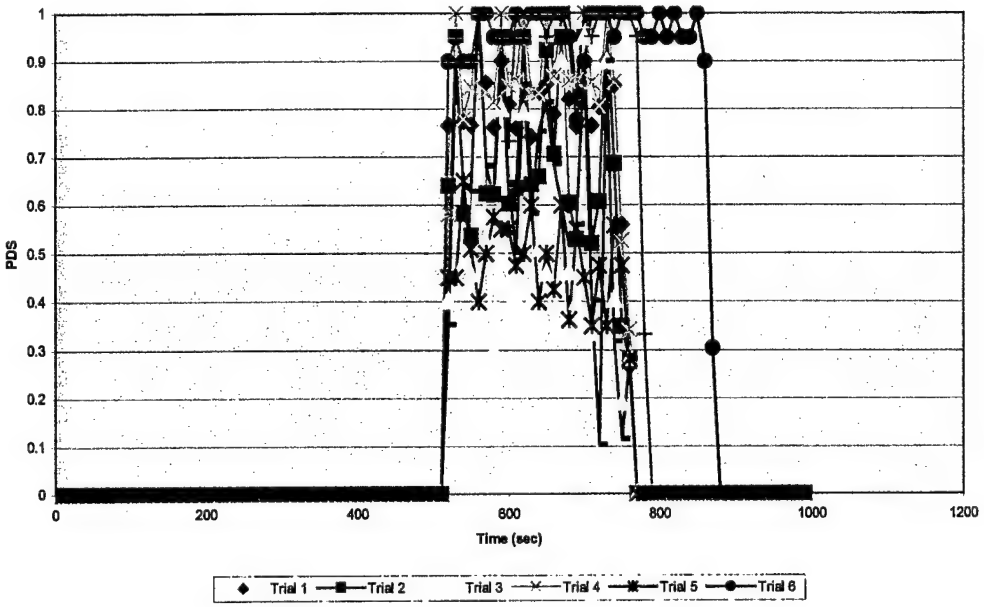

Figure 5-2 Average PDS as a Function of Time (TCP SYN Flood Attack)

Figure 5-3 presents an example of the packets forwarded by the router used by the attacker. The total rate of packets generated by the zombies exceeds the maximum packet-forwarding rate of the router. As such, the router's internal buffer fills with attack packets. When the attack ceases, the router continues to forward attack packets to the victim. This results in an extension of attack effects beyond the time it actually ends.

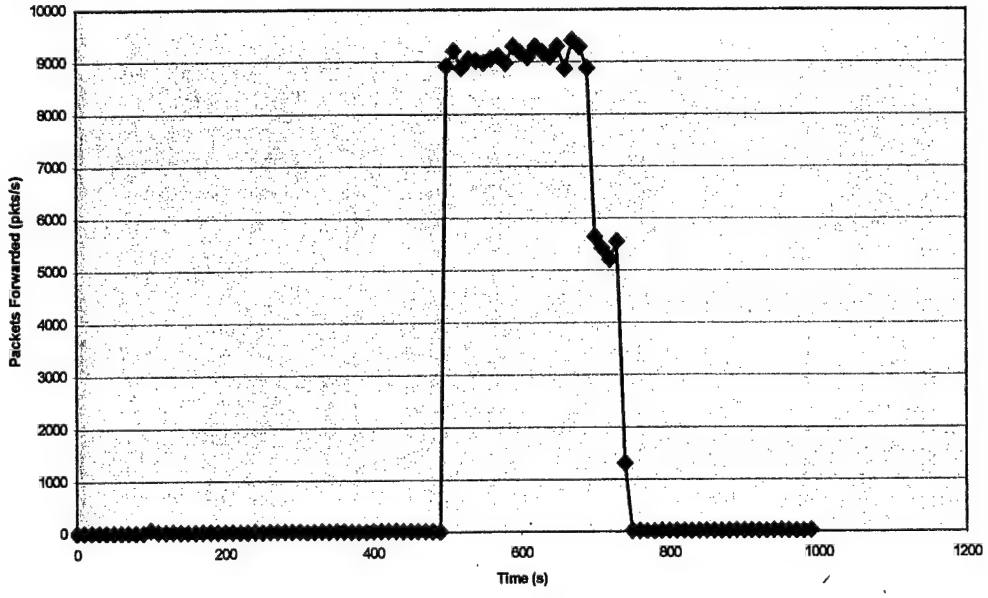

Figure 5-3 Router Packet Forwarding (TCP SYN Flood Attack)

5.3.2 VARYING THE QUEUE TIMEOUT

As stated previously, TCP SYN Flood attack effects depend on the queue timeout. The timeout parameter determines how long a pending connection is kept in the connection queue before being dropped. From a user's perspective, a shorter timeout means that resources are released more frequently and therefore more opportunities exist to gain service. This analysis examines three different timeout values: 15s, 90s, and 180s. Table 5-2 summarizes PDS for the three timeout values before, during, and after the attack.

Table 5-2 Average PDS as a Function of Queue Timeout (TCP SYN Flood Attack)

	PDS		
Timeout (sec)	Before	During	After
15	0	0.88	0.16
90	0	0.98	0.25
180	0	0.99	0.52

As expected, there is no DoS before the attack begins. Once the attack begins, the PDS increases as the timeout increases. Because a longer timeout means that resources are held by the attacker for a longer period of time, this is expected. Finally, there is a continuation of DoS after the attack has completed, and the longer timeout cases show an increased PDS over the 300-second window following the attack. Because the longer timeout cases hold resources longer into the post-attack period, a higher PDS occurs.

Longer timeout values also affect the duration of attack effects. Table 5-3 compares the length of the attack's impact as a function of timeout. Table 5-3 shows the time (in seconds) to a 10-second window when no DoS occurs. In the 15-second case, the duration of the attack lasts 70 seconds beyond the attack's completion. In this case, attack packets, queued at the router after the attack has ended, continue to create DoS effects beyond the attack's completion. For the 90- and 180-second cases, queuing at the router still occurs, but completes in the same 70-second timeframe. Thus, the attack duration is not extended beyond the 90- and 180-second periods.

Table 5-3 Attack Duration as a Function of Queue Timeout (TCP SYN Flood Attack)

Timeout (sec)	PDS Duration
15	70
90	90
180	180

Varying the queue timeout had no effect on services other than HTTP (e.g., the FTP service present on the other servers). That is, attack effects for these cases are isolated to the HTTP service under attack. As will be discussed, this is not always the case.

5.3.3 VARYING THE QUEUE SIZE

Three different queue values were selected for this portion of the assessment: 1024 (representative of the queue size in several versions of Solaris), 4096, and 8192. Table 5-4 shows PDS values for the three queue cases. PDS decreases as the queue size increases during and after the attack. As the queue size increases, two things happen: (1) the attacker must work harder to maintain a full queue and (2) the number of opportunities for the legitimate user to use the system increases.

Table 5-4 Average PDS as a Function of Queue Size (TCP SYN Flood Attack)

Queue Size	PDS		
	Before	During	After
1024	0.00	0.88	0.16
4096	0.00	0.74	0.11
8192	0.00	0.52	0.07

The modification of queue size causes no noticeable effect on the duration of the attack, unlike the effect seen when the timeout parameter is varied. However, it does affect the FTP service running on a server not under attack but existing on the same subnet. Figure 5-4 displays the FTP service time, as measured at the server.

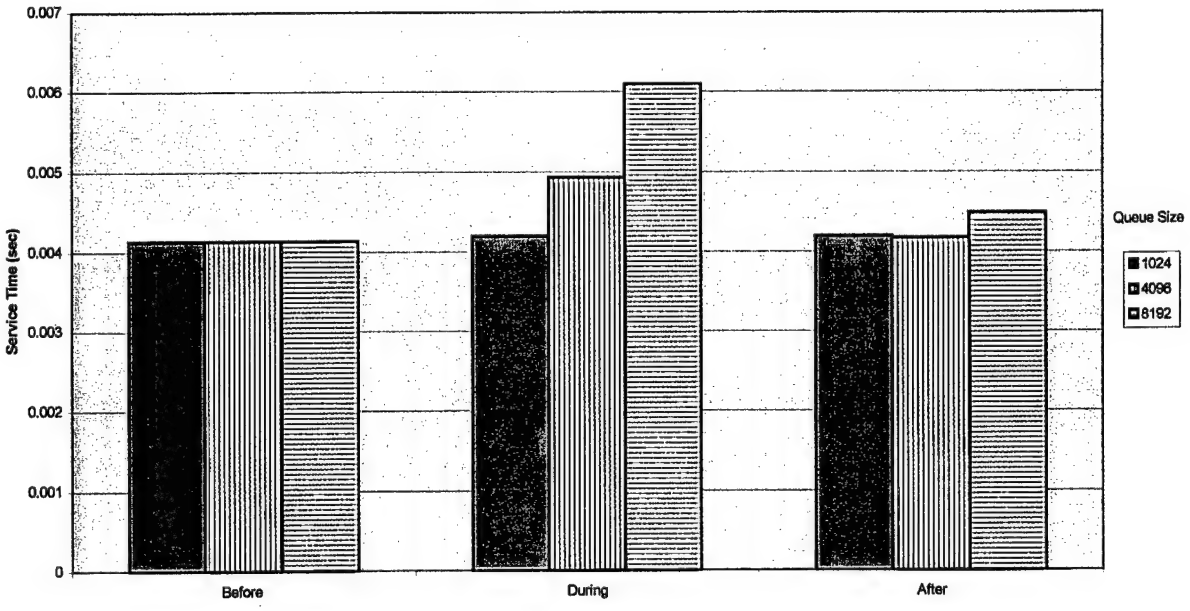

Figure 5-4 FTP Service Time as a Function of Queue Size (TCP SYN Flood Attack)

To explain the increase shown in Figure 5-4, Figure 5-5 presents example data for the packets forwarded as a function of time for the local router node and shows an increase in packet flows for the larger TCP queue. This is because of the increased traffic that occurs with lower PDS and the increase in traffic because of the increased number of SYN-ACK packets (a larger queue means more pending connections). This suggests more traffic on the network causing larger delays resulting in longer service times. Therefore, while this mitigation technique will reduce the effectiveness of the attack, it also has an impact on the larger network infrastructure.

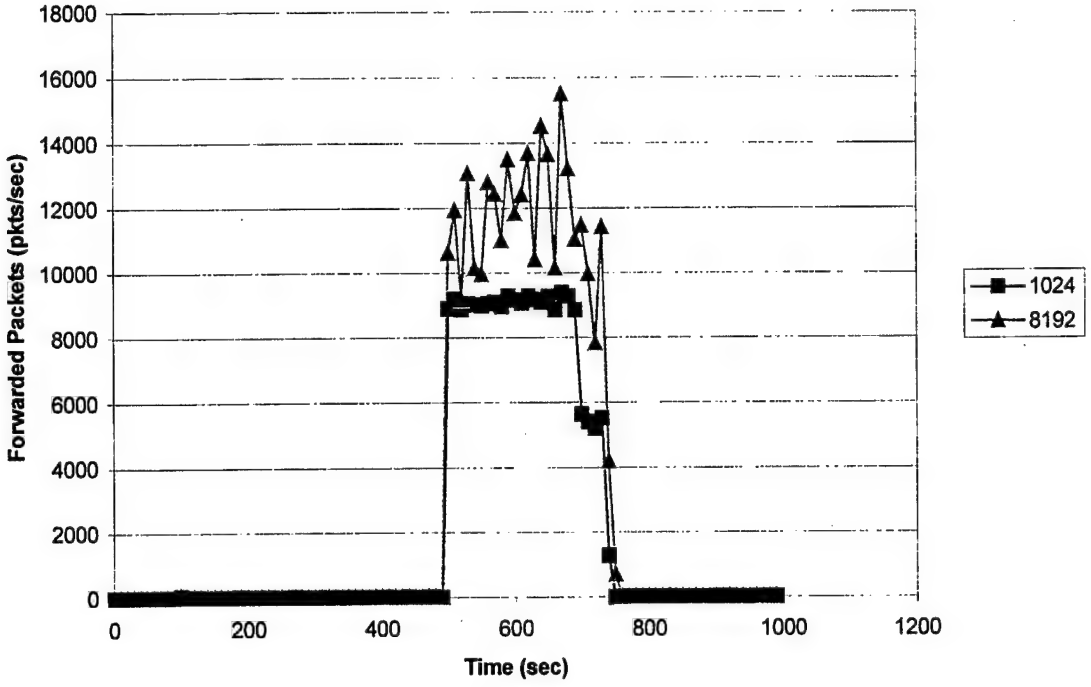

Figure 5-5 Router Packet Forwarding (TCP SYN Flood Attack)

5.3.4 VARYING SYN RATE

The final analysis for the TCP SYN Flood attack examines the impact of increasing the attack rate by directly injecting SYN packets at a rate of 5000 packets per second into the switch used by the two servers. This additional input, combined with the existing attack, results in a total of 10,000 packets per second. Table 5-5 summarizes PDS values for the original 5000-packets-per-second rate and the 10,000-packets-per-second case. The queue size for both cases is 8192. As expected, the attacker is able to substantially increase PDS by sending more packets.

Table 5-5 Comparison of Average PDS as a Function of Attack Rate (TCP SYN Flood Attack)

Aggregate SYN Rate (packets per second)	PDS		
	Before	During	After
5000	0	0.52	0.07
10,000	0	0.98	0.07

6.2 SNORK ANALYSIS RESULTS

Figure 6-1 presents sample results for one of the hosts in the subnet. As expected, the required download time increases during the attack but remains relatively unchanged during periods when the attack is not active. As the delay between the attack packets increases, the delay in downloading packets decreases. Although these effects are measurable, they are minor considering the timeframes involved. (The maximum difference is less than 1%.) However, the loading on the system in question prior to the attack is less than 0.5%. It is not surprising that these requests are not significantly affected by the Snork attack because they require minor CPU resources.

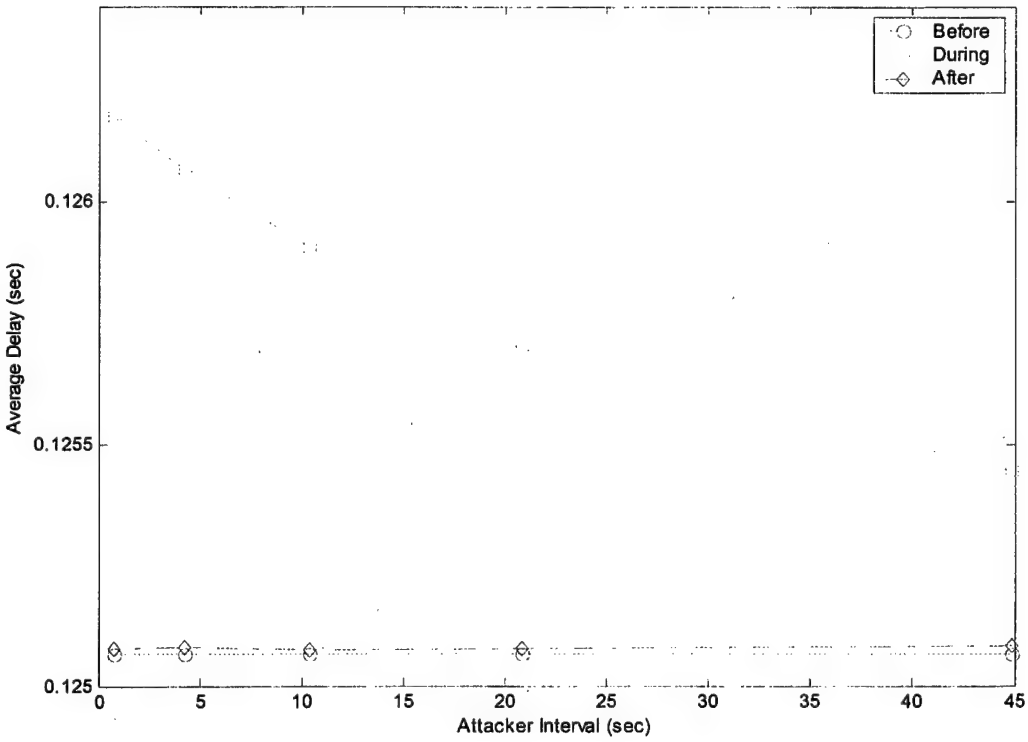

Figure 6-1 Average Download Delay (Snork Attack)

To further analyze Snork, the sizes of Web page objects were increased by a factor of 100 to determine Snork attack effects on a more heavily loaded system. The increased packet sizes resulted in a CPU load of approximately 15%. Figure 6-2 compares the average download time before, during, and after the Snork attack. As expected, there is very little difference between the times before and after the attack, but the delay increases during the attack. The greatest increase is observed when the attacker transmits packets at the quickest rate. Note that the increase in delay is greater, in terms of the percentage of the base value, for the more heavily loaded system (i.e., Figure 6-1 compared with Figure 6-2).

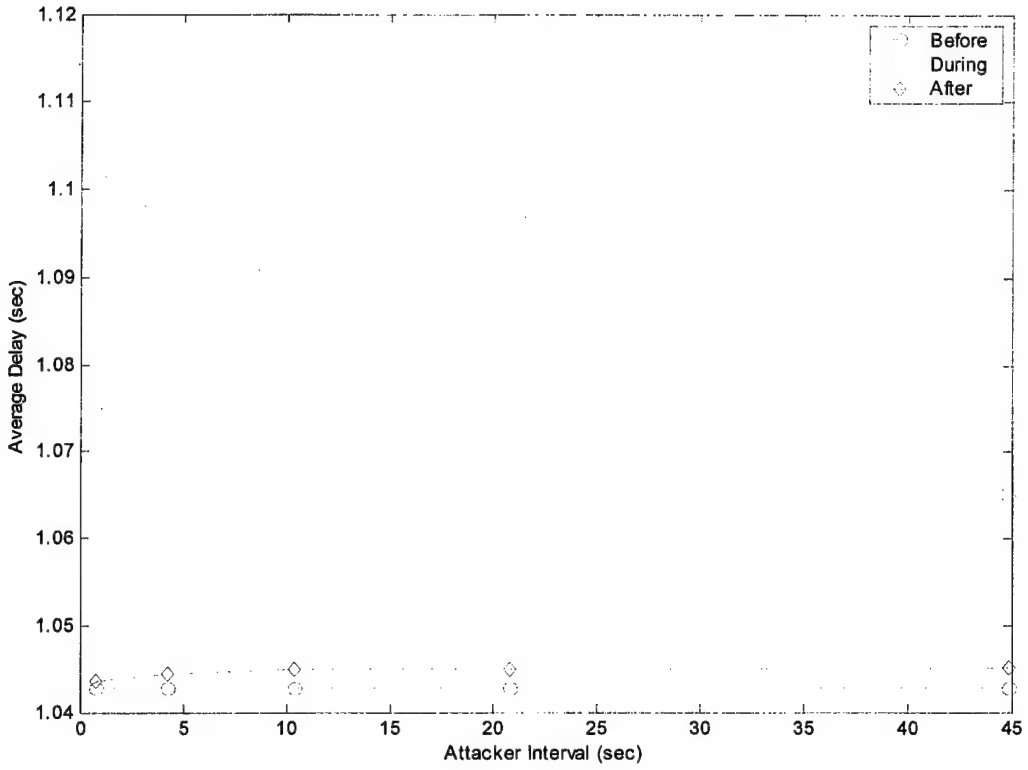

Figure 6-2 Average Download Delay for Larger Web Objects (Snork Attack)

As a final analysis, the client HTTP request generation rate was increased by changing the mean interarrival time from 60 seconds to 10 seconds, resulting in a CPU load of approximately 80%. The Snork attacker was configured to generate attack packets every 0.8 second. Under these conditions, the average delay before and after the attack is approximately 1.25 seconds, while during the attack it is 1.44 seconds, an increase of 15%.

This analysis suggests that because the CPU resource can be shared (i.e., the multi-tasking capabilities of the system allow multiple processes to operate at the same time), an attacker cannot deny service by using resources. However, the quality of service can be degraded: the amount of degradation, depending on the existing use of that resource and how much the attacker consumes.

Section 7

UDP STORM ATTACK

This section provides results from the analysis of a UDP Storm attack. The UDP Storm attack consumes network resources in an effort to deny or degrade the performance of the target network. To do this, the attacker generates a packet that is sent to a host running a service that will respond to the packet (e.g., echo). The service responds to the supposed source of the packet, which the attacker has spoofed to be echo process of another host on the network. When this packet is received by the spoofed host's process, it responds to the first host. This creates a continuous stream of packets between the two nodes. The attacker can then consume more resources if the attacker generates additional packets, thereby inducing more streams of data (either between an existing set of hosts or between different host pairs).

7.1 NETWORK CONFIGURATION

The UDP Storm attack was studied using the 104-node network shown in Figure 7-1. All hosts are connected to the LAN by a switch. In addition, the subnet was modified to place more stress on the network (similar to the increase in CPU loading for the Snork analysis). Specifically, traffic throughout the network was increased, and the switch processing speed and queue size were decreased.

7.2 ANALYSIS PARAMETERS

As stated, UDP Flood operates by creating a continuous stream of packets between two host systems with a goal of denying service by overwhelming network resources. Three potential resources could be overwhelmed:

a. Data communications link between individual hosts and the switch

b. Hosts involved in the flooding

c. Network switch

Given a 100-Mbps switched LAN topology (used in the 104-node network) and a maximum packet size of 1500 bytes, a host would be required to forward 8333 packets per second to fully use the data link. Because the hosts used in the model only forward a maximum of 5000 packets per second, the configuration used in the model will not permit the data link to be exhausted.[1]

[1] This will be examined in a DDoS architecture during the next year of this project.

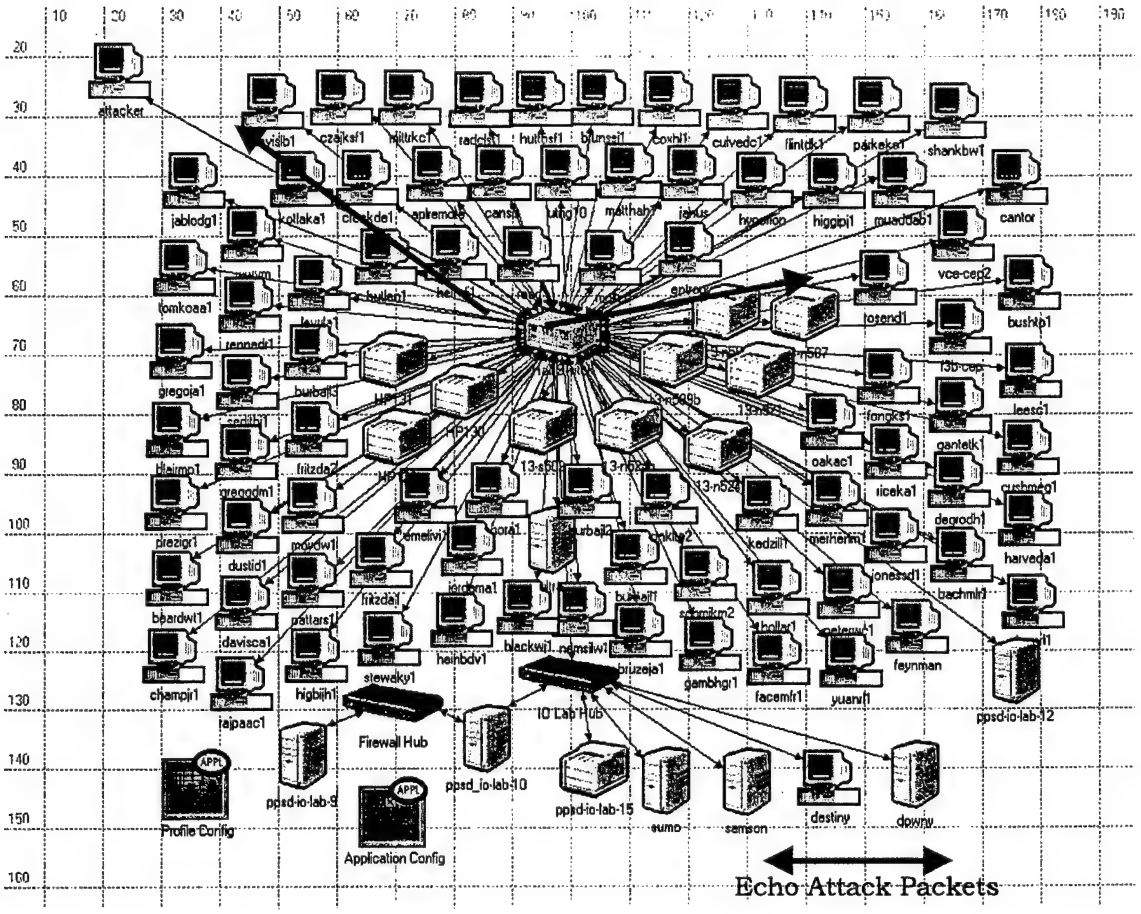

Figure 7-1 Network Used for UDP Storm Attack

The second option is to create sufficient packets between two hosts to cause a backlog of packets between the two nodes. If the attacker could induce an exchange in excess of 5000 packets per second, a backup of packets within a host or set of hosts could occur. In addition, given finite buffering in each host, packets sent by the host could be lost. In this case, if the switch has sufficient capacity, other clients on the network would not be affected.

The third option explored in this study is to use multiple connection pairs to overwhelm the switch, thereby causing degradation across the network. In this case, the goal is to overwhelm the switch's capability to switch packets across the network by using multiple pairings. This would result in dropped packets at the switch and delays for all hosts on the network. To determine how network parameters influence the effectiveness of the attack, three parameters are varied:

a. Number of host pairs attacked

b. Switch queue size in kilobits

c. Speed of the switch in kilopackets per second (kpps)

The parameter values for each model run are shown in Table 7-1.

Table 7-1 UDP Storm Attack Analysis Cases

Case	No. of Pairs	Switch Speed (kpps)	Switch Queue Size (kb)
1	0	10	120
2	2	10	120
3	4	10	120
4	8	10	120
5	10	10	120
6	10	10	150
7	10	10	200
8	10	10	500
9	10	20	120

In all cases, the attacker continued to generate a new UDP Storm packet every 5 seconds during the attack period (500 to 1000 seconds). This is to ensure that any packets dropped by the switch would be replaced and that the system was nominally under attack conditions.

7.3 UDP STORM ANALYSIS RESULTS

The primary data collected from the model is the observed data throughput at two clients within the subnet. One of these clients is actively participating in the attack and is referred to as the *active* client. The second is a system attached to the switch but is not directly subject to the UDP Storm attack. This system is referred to as the *passive* client. Figure 7-2 presents the data throughput for various combinations of attack pairings and for a baseline case when no attacks were present.

Figure 7-2 shows that both the active and passive clients' throughput decreases while under attack but significant changes do not occur as the number of pairings increase. Because the attacker is constantly generating new attack packets, the network is operating at capacity, regardless of how many attacker pairs are present in the system.

Figure 7-3 shows the impact of changing the queue size on the attack. Because the attacker is continuously transmitting additional attack packets, the queue is filling to capacity in each case. The addition of queue resources does not significantly change the performance of the system.

Figure 7-4 summarizes model results as a function of switch speed. As the speed of the switch is increased, the throughput of the active and passive clients increases. The faster switch compensates for the network load imposed by UDP Storm.

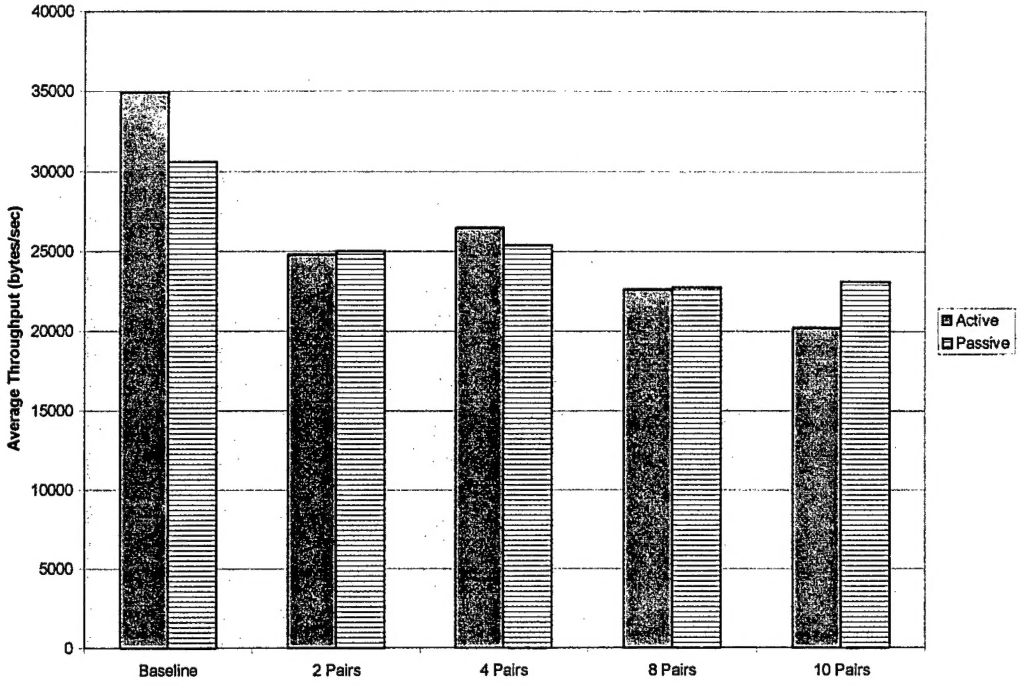

Figure 7-2 Throughput as a Function of Attack Pairings (UDP Flood Attack)

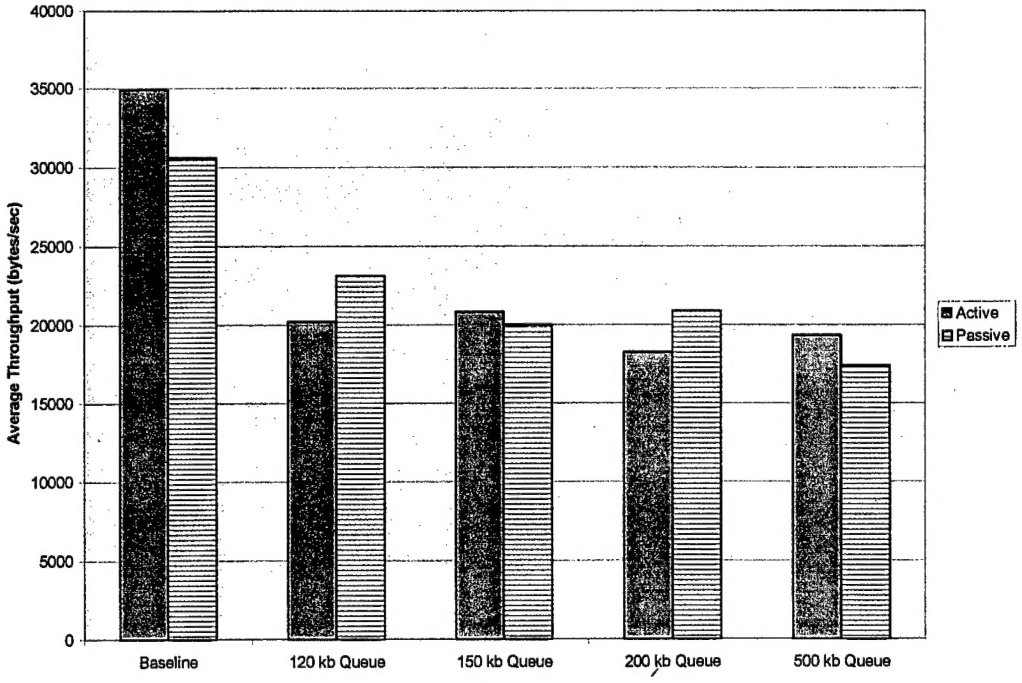

Figure 7-3 Throughput as a Function of Queue Size (UDP Flood Attack)

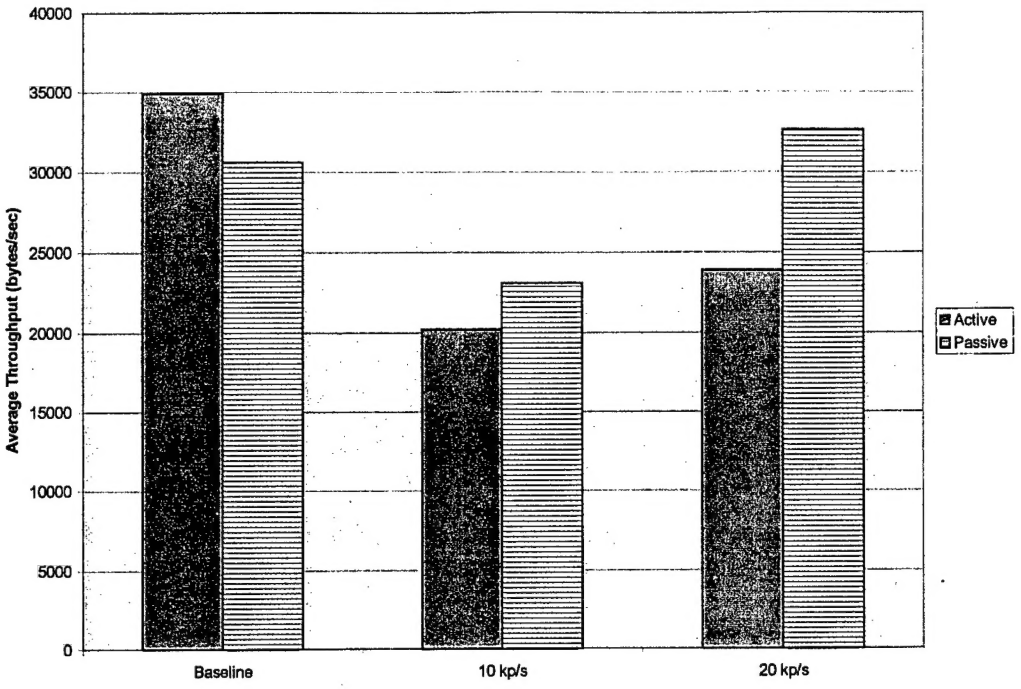

Figure 7-4 Throughput as a Function of Switch Speed (UDP Flood Attack)

In addition to average data throughput, model output data was also processed to determine if service was denied at any point. Model data indicates that while service was degraded by the UDP Storm attack, no services were denied. Specifically, no instances of connection failure (e.g., due to dropped packets) were observed.

Section 8

CONCLUSIONS AND OBSERVATIONS

DOSAA work has shown that modeling and simulation can be used to quantify DoS attacks, mitigation techniques, and the attacker's ability to adapt. The following observations are made regarding this analysis:

a. Attacks do fall into attack classes. One class characterized in this analysis is the stateful resource attack of which Octopus and TCP SYN Flood are members. This analysis has revealed that these attacks are driven by very specific parameters: the amount of resource to be consumed, the length of time the system holds the resources before releasing them, and the rate at which the attacker consumes resources. Also, all things being considered equal between the client and attacker, the effectiveness of these attacks is independent of network configuration.

b. Some attacks are highly dependent on the victim network's topology. In the ARP Cache Poison attack, PDS varied greatly between the two networks examined. Because this attack is based on a race condition, network delay (and hence topology) drives attack effectiveness.

c. It may be more difficult for the attacker to deny service when resources are shared. In the Snork and UDP Storm attacks, the CPU resource and channel, respectively, are shared, and no DoS was observed in this analysis. However, the quality of service can be degraded: the amount of degradation, depending on the existing use of that resource and how much the attacker consumes.

d. Attacks require varying degree of work to execute. In this analysis, the ARP Cache Poison and Snork attacks take one packet to execute, Octopus and TCP SYN Flood attacks require hundreds of packets, and UDP Storm takes thousands of packets to achieve the desired effect.

MISSION
OF
AFRL/INFORMATION DIRECTORATE (IF)

The advancement and application of Information Systems Science and Technology to meet Air Force unique requirements for Information Dominance and its transition to aerospace systems to meet Air Force needs.